PSG2

PROCESO SOFTWARE Y GESTIÓN II

Informe técnico “métricas de proceso ágil y recursos”

product owner: CARLOS MÜLLER CEJÁS

Carmen Mª Muñoz Pérez (Scrum Master)

Enrique Salazar Márquez

Javier Martínez Fernández

José Carlos Morales Borreguero

Rafael Ángel Jiménez Fernández

Contenido

[Informe técnico del Sprint 2 2](#_Toc70530283)

[Sprint 2: Gráfica “Burndown” 2](#_Toc70530284)

[Sprint 2: Total de puntos de historia entregados. 2](#_Toc70530285)

[Sprint 2: Porcentaje de puntos de historia entregados 2](#_Toc70530286)

[Informe técnico del Sprint 3 3](#_Toc70530287)

[Sprint 3: Gráfica “Burndown” 3](#_Toc70530288)

[Sprint 3: Gráfico de control (Lead Time) 3](#_Toc70530289)

[Sprint 3: Gráfico de control (Cycle Time) 4](#_Toc70530290)

[Sprint 3: Total de puntos de historia entregados. 4](#_Toc70530291)

[Sprint 3: Porcentaje de puntos de historia entregados 5](#_Toc70530292)

[Sprint 3: Calendario Niko-Niko 5](#_Toc70530293)

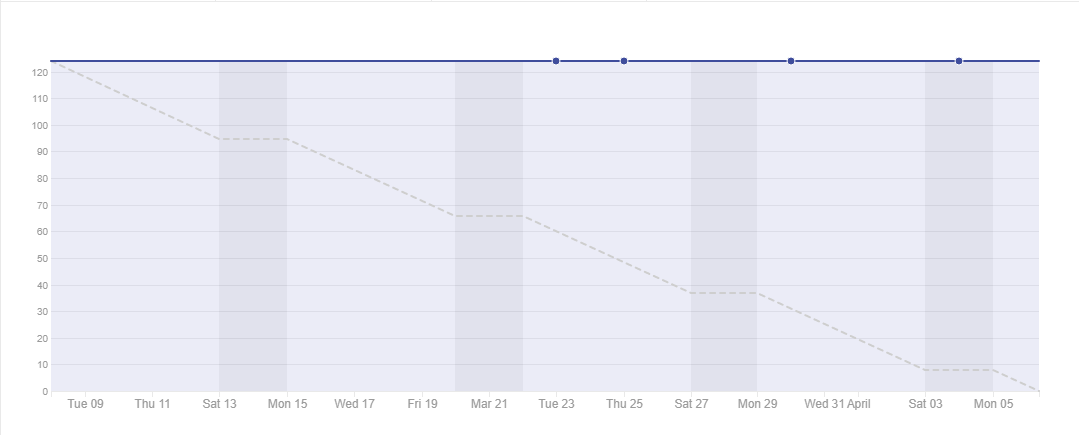
[E3.3: Apéndice política de ramas 7](#_Toc70530294)

[Integración del uso de épicas en la metodología de ramificación. 7](#_Toc70530295)

### Informe técnico del Sprint 2

### Sprint 2: Gráfica “Burndown”

Nuestro Product Owner nos propuso utilizar “ZenHub” una vez comenzado el tercer sprint, por tanto, las métricas de este se han visto afectadas ya que no realizamos los pasos necesarios y en orden para que se pudieran ver reflejadas de forma coherente en los gráficos debido a nuestro desconocimiento previo.



### Sprint 2: Total de puntos de historia entregados.

Como se puede observar, entre todas las tareas sumaban 124 puntos de historia y los 124 han sido entregados a tiempo.

****

### Sprint 2: Porcentaje de puntos de historia entregados

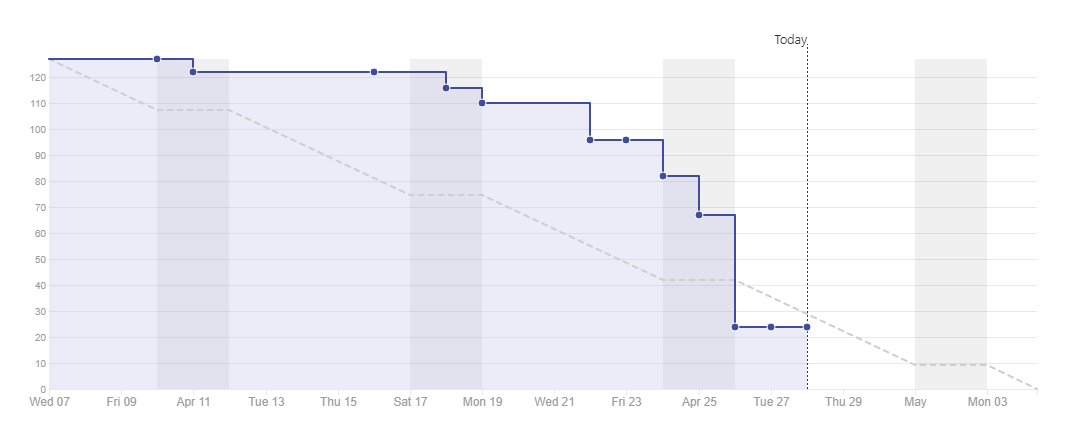
Como se puede observar bajo la barra de progreso, el sprint 2 fue un éxito y está al 100% de puntos de historia entregados. Se adjunta también un breve ejemplo de las tareas cerradas de dicho sprint.

****

### Informe técnico del Sprint 3

### Sprint 3: Gráfica “Burndown”

Esta sería la gráfica “Burndown” del tercer sprint a día de hoy. Como se puede observar, se han estado cumpliendo los plazos. Al estar trabajando en 5 tareas simultáneamente, se puede apreciar un escalón importante ya que algunas tareas dependían del avance de otras para estar completas. No obstante, el progreso es adecuado y los puntos han sido repartidos de manera correcta y precisa.

****

### Sprint 3: Gráfico de control (Lead Time)

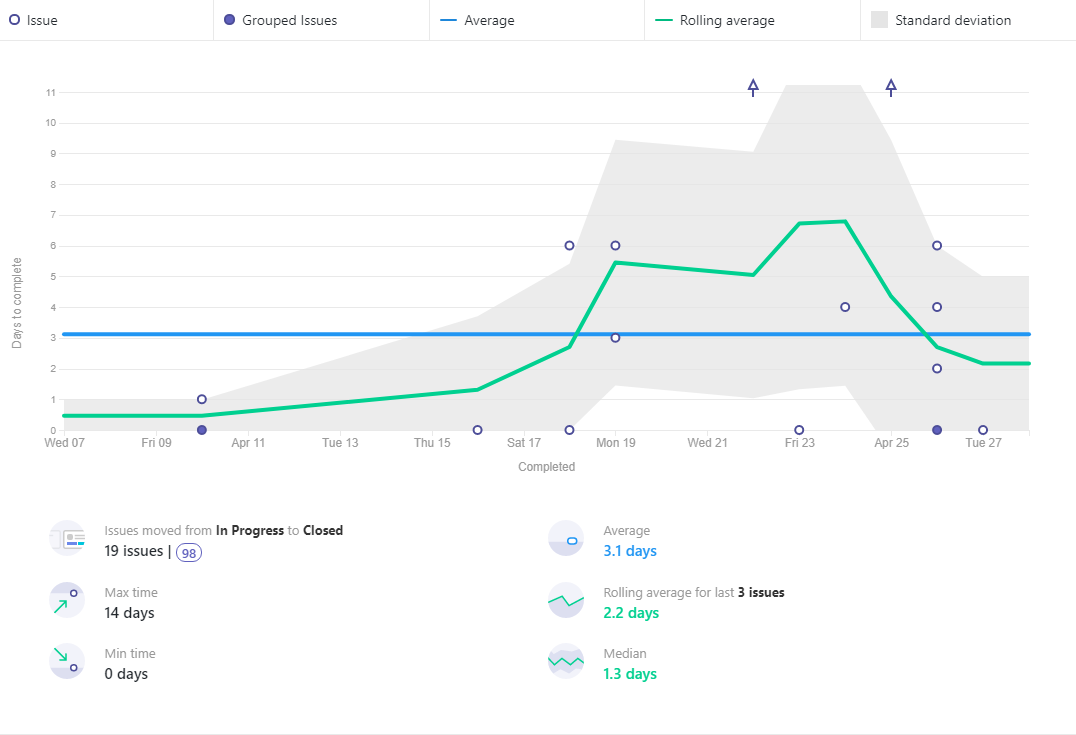
El siguiente gráfico muestra el tiempo total desde que se crea una tarea hasta que esta es completada, en este caso, el tiempo que tarda la tarea de pasar desde “New Issue” a “Closed” .



### Sprint 3: Gráfico de control (Cycle Time)

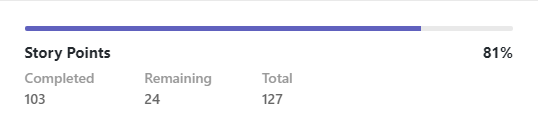
El siguiente gráfico muestra el tiempo del ciclo, es decir, el tiempo en el que se ha estado trabajando en una tarea activamente, en este caso, desde que la tarea pasa a estar en “In progress” a “Closed”.

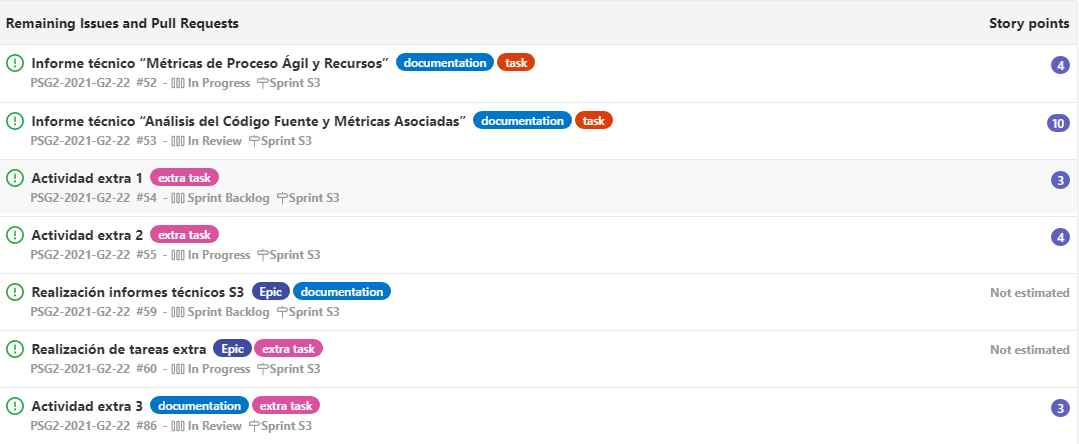
Cabe destacar que, en este Sprint al no estar acostumbrados a trabajar con épicas, no se han puesto en “In progress” al empezar a trabajar en ella, ya que sólo se iban poniendo en “In progress” las tareas correspondientes. Por ello en esta métrica aparecen con una duración de 0 días. En el siguiente Sprint lo haremos bien desde el principio.

****

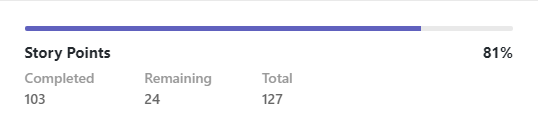
### Sprint 3: Total de puntos de historia entregados.

Como se puede observar, entre todas las tareas sumaban 127 puntos de historia y hasta la fecha han sido entregados 103. Los restantes serían los documentos de finalización de sprint y algunas actividades extras.



****

### Sprint 3: Porcentaje de puntos de historia entregados



### Sprint 3: Calendario Niko-Niko

Durante la duración del sprint, todos los miembros del proyecto han ido actualizando diariamente, según su situación frente al trabajo asignado, este calendario Niko-Niko. La experiencia en general utilizando por primera vez esta herramienta ha sido satisfactoria, ya que nos ha invitado individualmente a hacer un ejercicio de introspección y así ir mejorando día a día.

La gama de sentimientos encontrados en el calendario es variada, pero sobre todo predominan la “smiley face” y la “neutral face”. La primera es indicativa de un día de provecho y sin ningún estancamiento, la segunda es de un día satisfactorio pero con algún que otro problema con solución no muy costosa. Ha habido muy pocos “worried” y “fearful” ya que existía una muy buena comunicación entre todos los miembros del proyecto, y al mínimo problema se ha ayudado al compañero.

Para finalizar, se ha aprendido que cuando un compañero actualizaba este calendario, se debería mirar cuanto antes para poder ayudar a este si estaba estancado con su trabajo; acción que hemos llevado a cabo en la fase final del sprint como recomendación de nuestro Product Owner.

****

****

### E3.3: Apéndice política de ramas

### Integración del uso de épicas en la metodología de ramificación.

Durante el tercer sprint se ha mantenido la metodología de ramas utilizada en el segundo sprint, debido a que cuando el equipo de desarrollo fue informado sobre integración de las épicas en la metodología de ramas, ya se había comenzado el sprint con el sistema anterior por lo que tras una deliberación se llegó a la conclusión de que para el tercer sprint se mantendría el sistema de ramas sin épicas. De esta forma se evitarían ciertos problemas que pueden aparecer al aplicar una metodología distinta a mitad de un sprint.

No obstante, la metodología de ramas épicas nos parece un buen sistema de ramas por lo que nuestra idea es implementarlo en el siguiente sprint. La forma en la que aplicaremos este sistema de ramas épicas será el siguiente:

* Una tarea épica tendrá su propia rama siguiendo la sintaxis “epic/X-descripcion”, donde X es el número correspondiente a la id de la tarea de GitHub asignado automáticamente una vez se crea esta (este número está generalmente acompañado de una #. Ej. #52). Esta tarea épica será descompuesta en pequeñas tareas que también tendrán su propia rama siguiendo la sintaxis “feature/X-descripcion”.
* Cuando una feature sea finalizada, revisada y aprobada, el contenido de esta rama se verterá en la épica correspondiente asociada.

De esta forma nos aseguramos de que el contenido correspondiente a una funcionalidad está bien organizado y que todo el código correspondiente a dicha funcionalidad está en su rama épica.